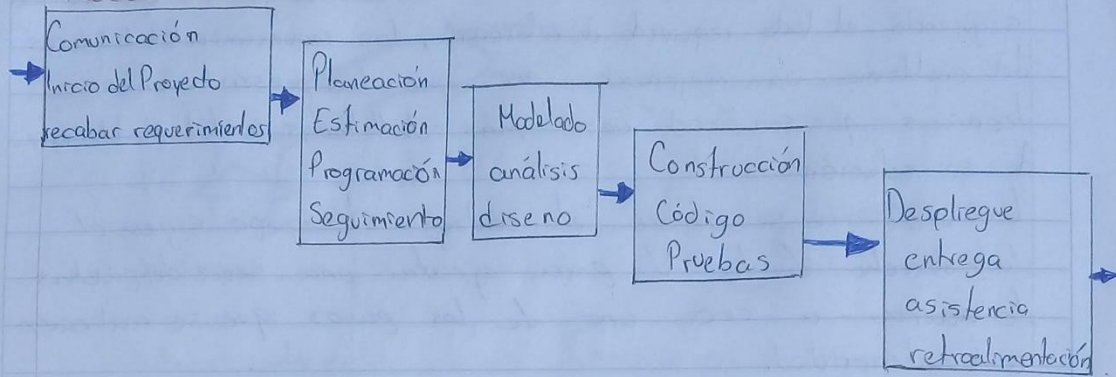


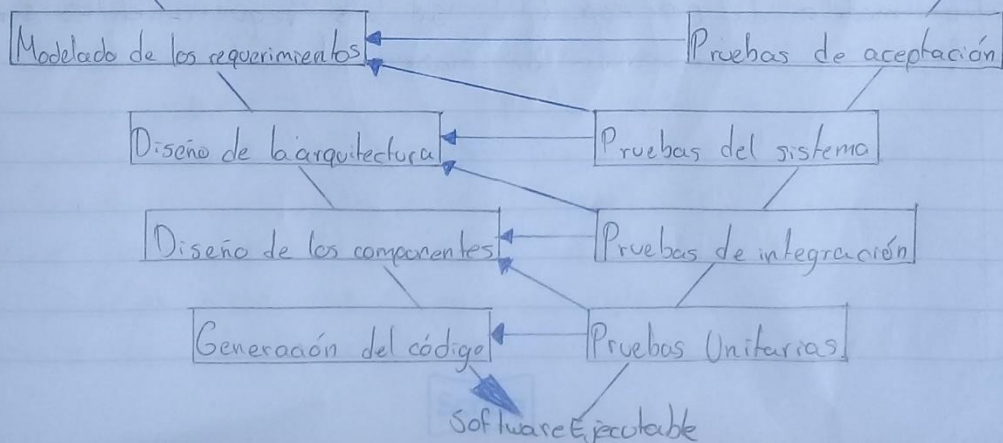
Actividad 17/11/2021

Modelo Cascada



También es llamado ciclo de vida clásico, todo el proceso que se realiza es secuencial, por lo general los proyectos reales no pueden seguir el flujo lineal del método aunque este acepte repeticiones y regresos a las fases anteriores, pero esto puede llegar a confundir a los integrantes del equipo.

En este modelo el cliente debe tener paciencia ya que no se dispone de versiones funcionales durante el proceso, hasta que este se finalice de manera correcta, una variante de este modelo se conoce como modelo V.



A medida que el equipo avance hacia abajo en este modelo siguiendo el lado izquierdo de referencia, los requerimientos básicos del problema van mejorando, así como las presentaciones técnicas que son cada vez más detalladas del problema y de su solución.

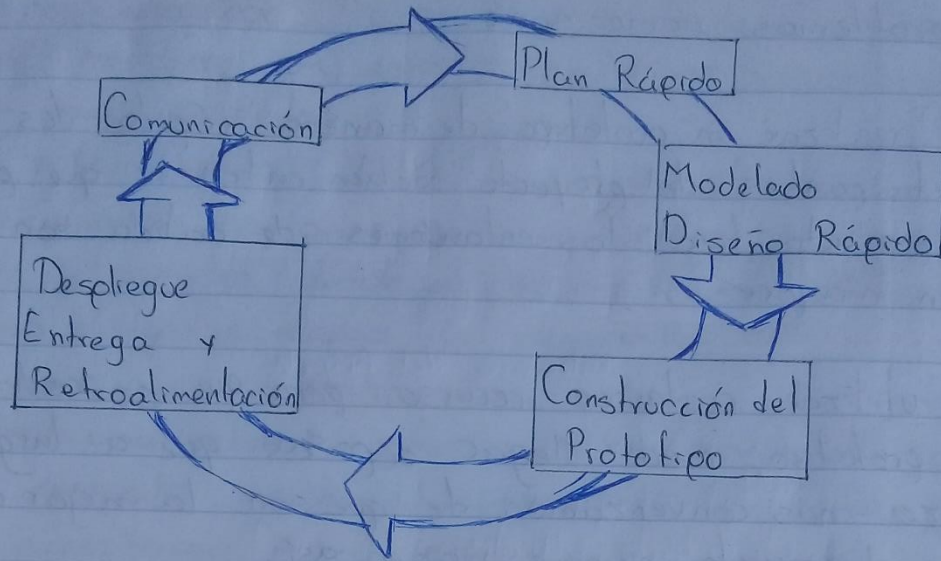
Una vez que se genere el código el equipo sigue el lado derecho de la V para ejecutar una serie de pruebas de acuerdo a cada uno de los pasos que se realizaron con anterioridad.

Este modelo es el más antiguo de la ingeniería de software pero presenta varios problemas algunos de ellos son:

1. Al no ser los proyectos de manera secuencial se pueden ocasionar confusiones
2. El cliente deberá esperar más tiempo para ver resultados sobre su proyecto.

Para el desarrollo de software en la actualidad, el modelo de cascada es inapropiado, pero, sirve como un modelo de proceso útil si los requerimientos son fijos y se avanza de manera lineal.

Modelos Evolutivos



Los modelos evolutivos son iterativos, se caracterizan por la manera en que permiten desarrollar versiones cada vez más complejas del software, algunos ejemplos son:

Hacer prototipos:

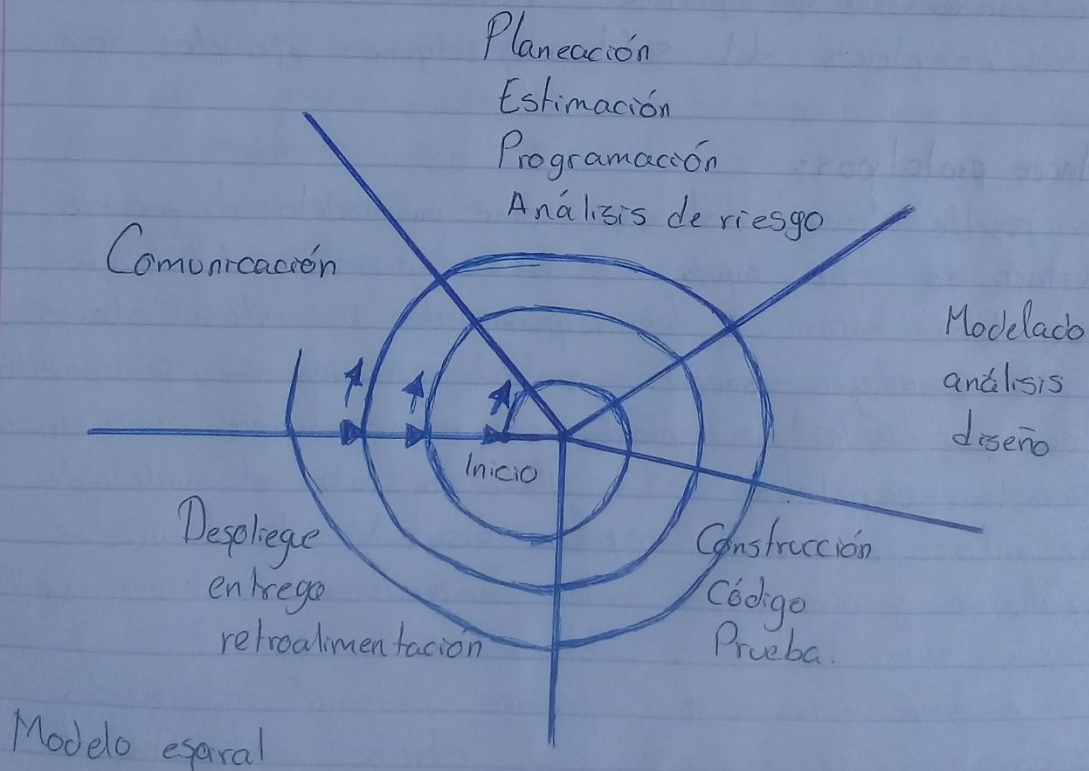
Es posible hacer prototipos como un modelo de proceso aislado, ya que ayuda a la comprensión de lo que se debe hacer si los requerimientos no están claros.

Este paradigma comienza con la comunicación, se identifican los requerimientos y se planea de manera rápida una iteración para hacer un prototipo y se lleva a cabo el modelado. Se enfoca en dar un aspecto visual de lo que se realizará y esto ayuda a la comprensión de lo que se desea hacer.

Un prototipo sirve como "primer sistema" en algunas ocasiones, se desecha y en otras evoluciona al

Producto final, pero este paradigma también cuenta con problemas como son:

1. Al realizar un prototipo de manera rápida los participantes del proyecto deben entender que para el producto final los estándares de calidad son aún mayores.
2. Al utilizar cualquier recurso para que se realice el prototipo, puede llegar a pasar que a largo plazo nos convenzamos de que son la mejor opción para el proyecto aunque no sea así.



Modelo espiral.

Fue propuesto por Barry Boehm, tiene el potencial de hacer prototipos con aspectos controlados y sistémicos, cada vez más completos.

Con este modelo el software que desarrolla va evolucionando conforme pasa el tiempo, se comienza desde el centro dando el mismo tiempo de prioridad a las distintas etapas del desarrollo.

Una vez que se realiza una entrega al cliente se realiza una retroalimentación y es donde el gerente del proyecto ajusta el número de iteraciones que se necesitarán para terminar el proyecto.

A diferencia de otros modelos, cuando se finaliza el proyecto el modelo puede seguir ocupándose para que el software pueda seguir creciendo y evolucionando, esto refleja de manera más realista a diferencia del cascada clásico que era lineal.

Desventajas

Es difícil convencer a los clientes que el modelo evolutivo es controlable.

Demanda experiencia en la evaluación de riesgo y se basa en ella para llegar al éxito.